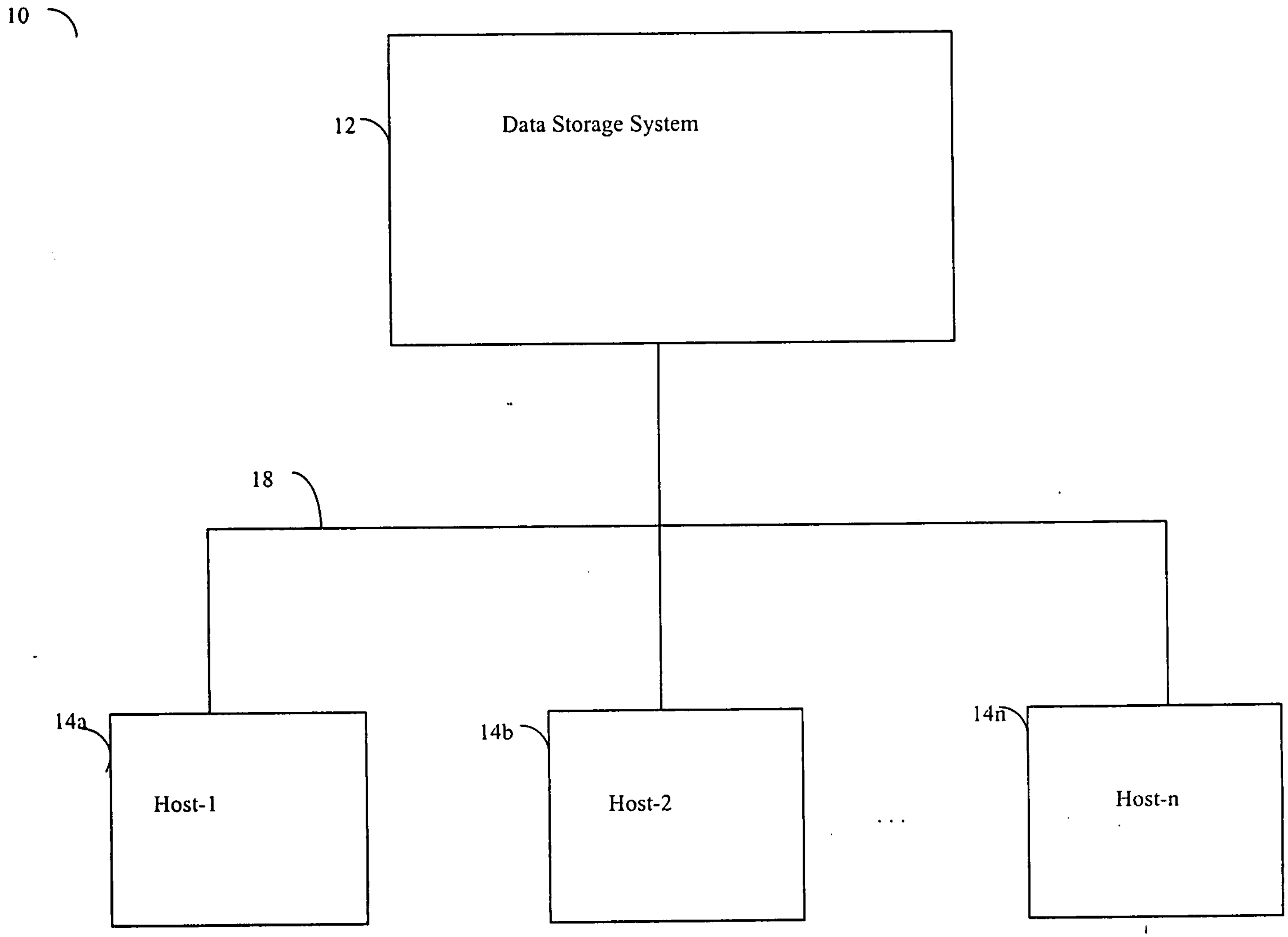


FIGURE 1

19

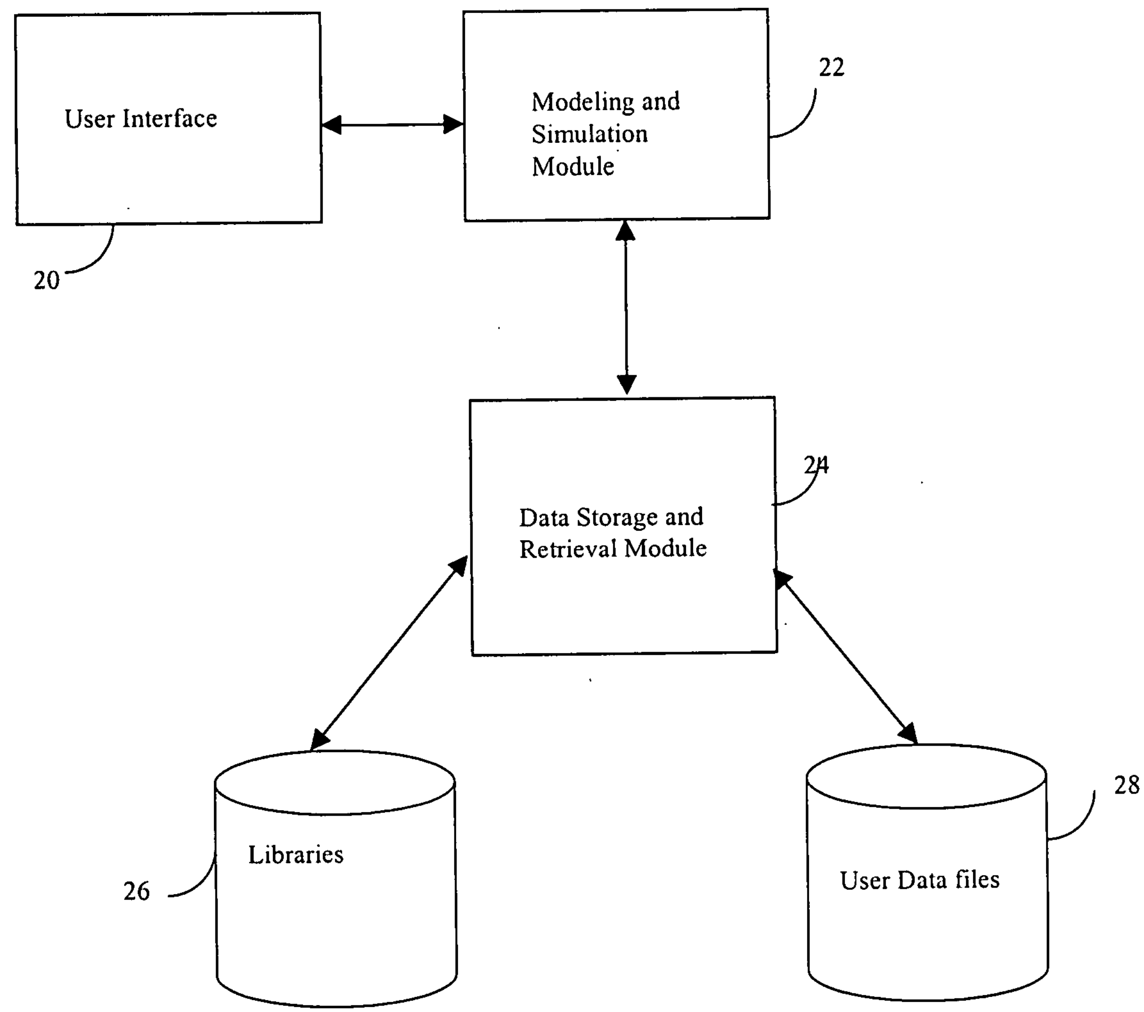
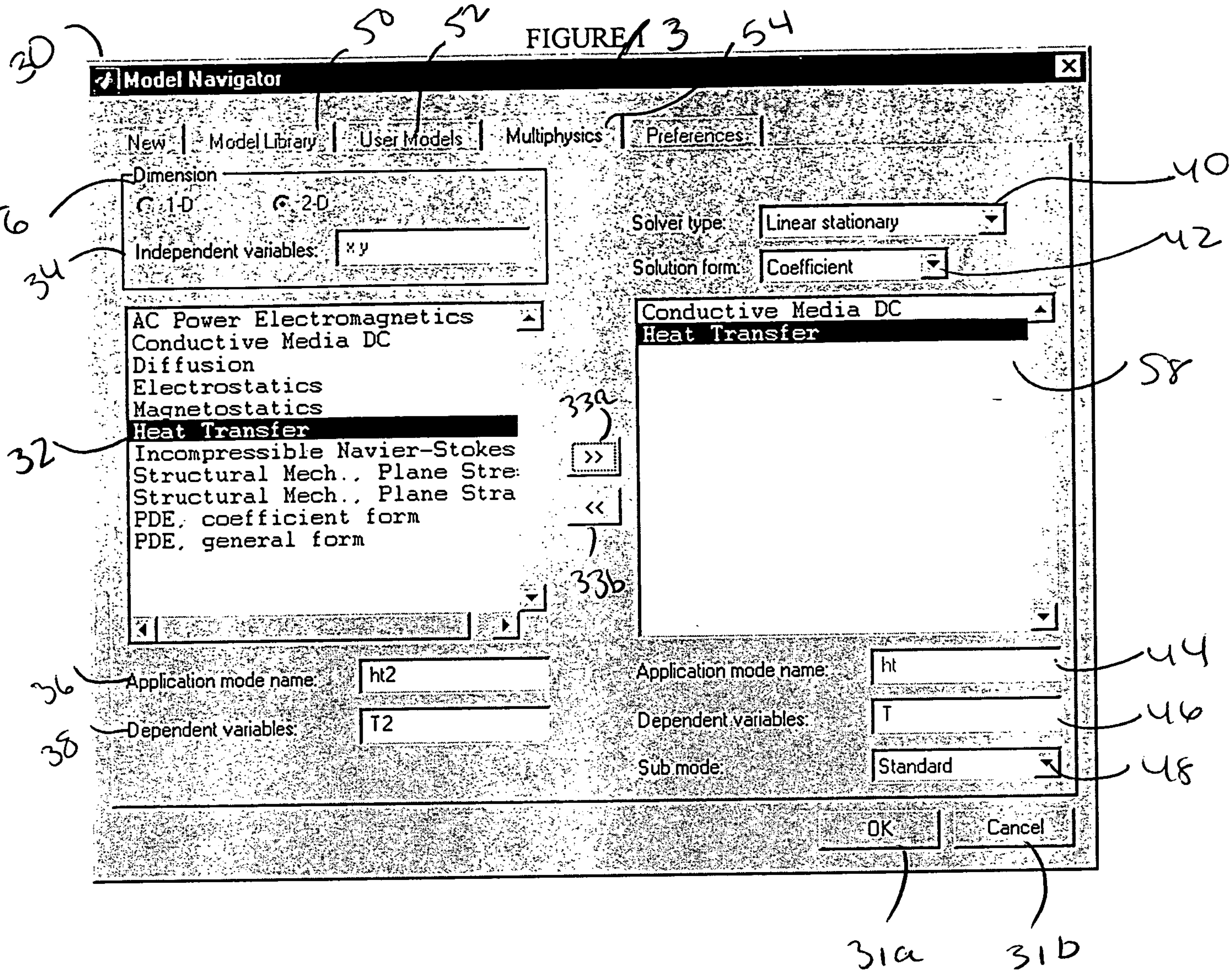

User Interface
20
Modeling and Simulation Module
22
Data Storage and Retrieval Module
24
Libraries
26
User Data files
28

FIGURE 2

FIGURE 3



30
50
52
54
Model Navigator
New
Model Library
User Models
Multiphysics
Preferences
Dimension
1-D
2-D
Independent variables:
x y
56
34
Solver type:
Linear stationary
40
Solution form:
Coefficient
42
AC Power Electromagnetics
Conductive Media DC
Diffusion
Electrostatics
Magnetostatics
Heat Transfer
Incompressible Navier-Stokes
Structural Mech., Plane Stre
Structural Mech., Plane Stra
PDE, coefficient form
PDE, general form
32
33a
33b
Conductive Media DC
Heat Transfer
58
Application mode name:
ht2
36
Dependent variables:
T2
38
Application mode name:
ht
44
Dependent variables:
T
46
Sub mode:
Standard
48
OK
Cancel
31a
31b

FIGURE ~~2~~ 4

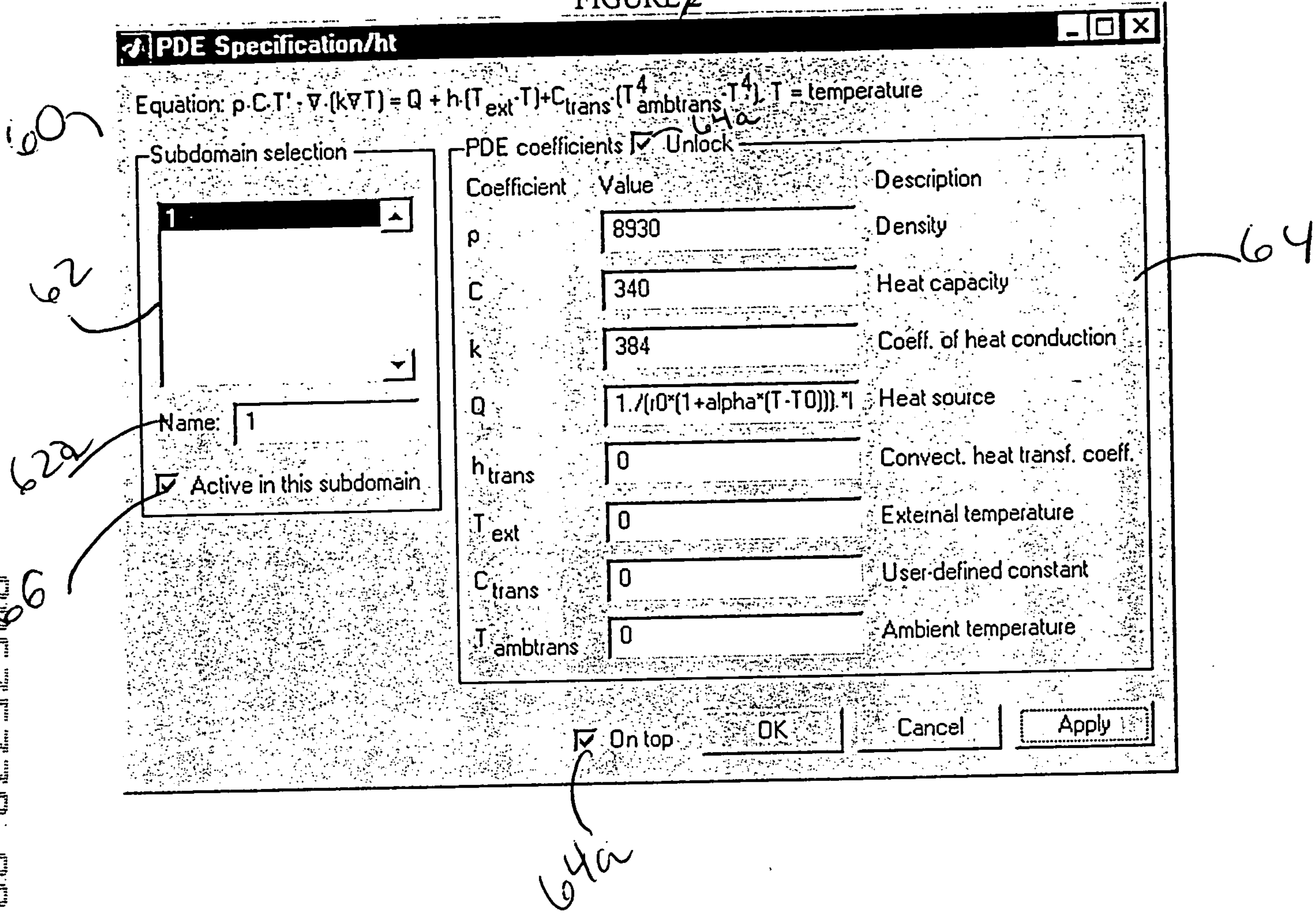
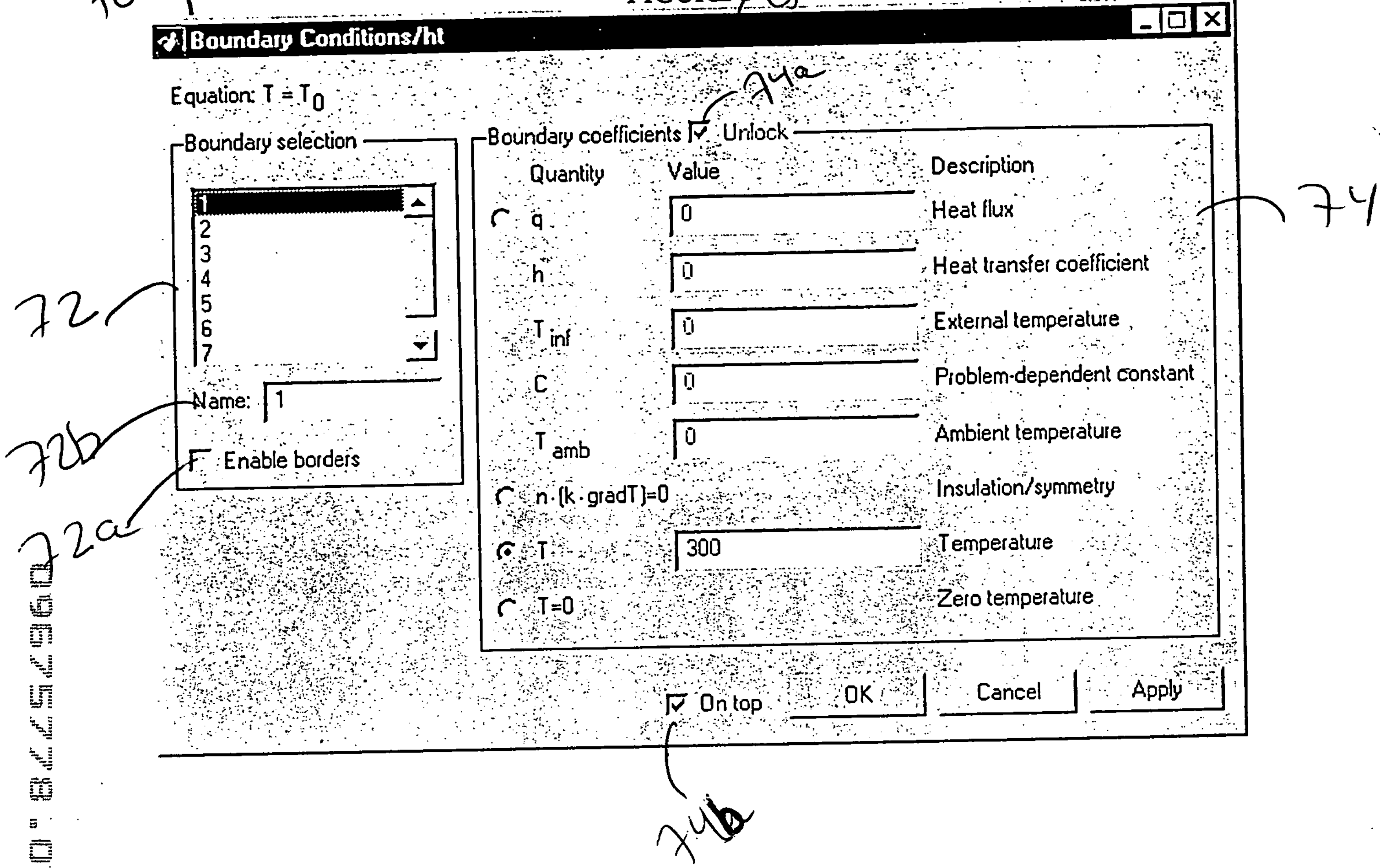

PDE Specification/ht
Equation: ρ·C·T' - ∇·(k∇T) = Q + h·(T_ext - T) + C_trans·(T^4_ambtrans - T^4), T = temperature
Subdomain selection
1
Name: 1
Active in this subdomain
PDE coefficients
Unlock
Coefficient
Value
Description
ρ
8930
Density
C
340
Heat capacity
k
384
Coeff. of heat conduction
Q
1./(r0*(1+alpha*(T-T0))).*I
Heat source
h_trans
0
Convect. heat transf. coeff.
T_ext
0
External temperature
C_trans
0
User-defined constant
T_ambtrans
0
Ambient temperature
On top
OK
Cancel
Apply
60
62
62a
64
64a
66

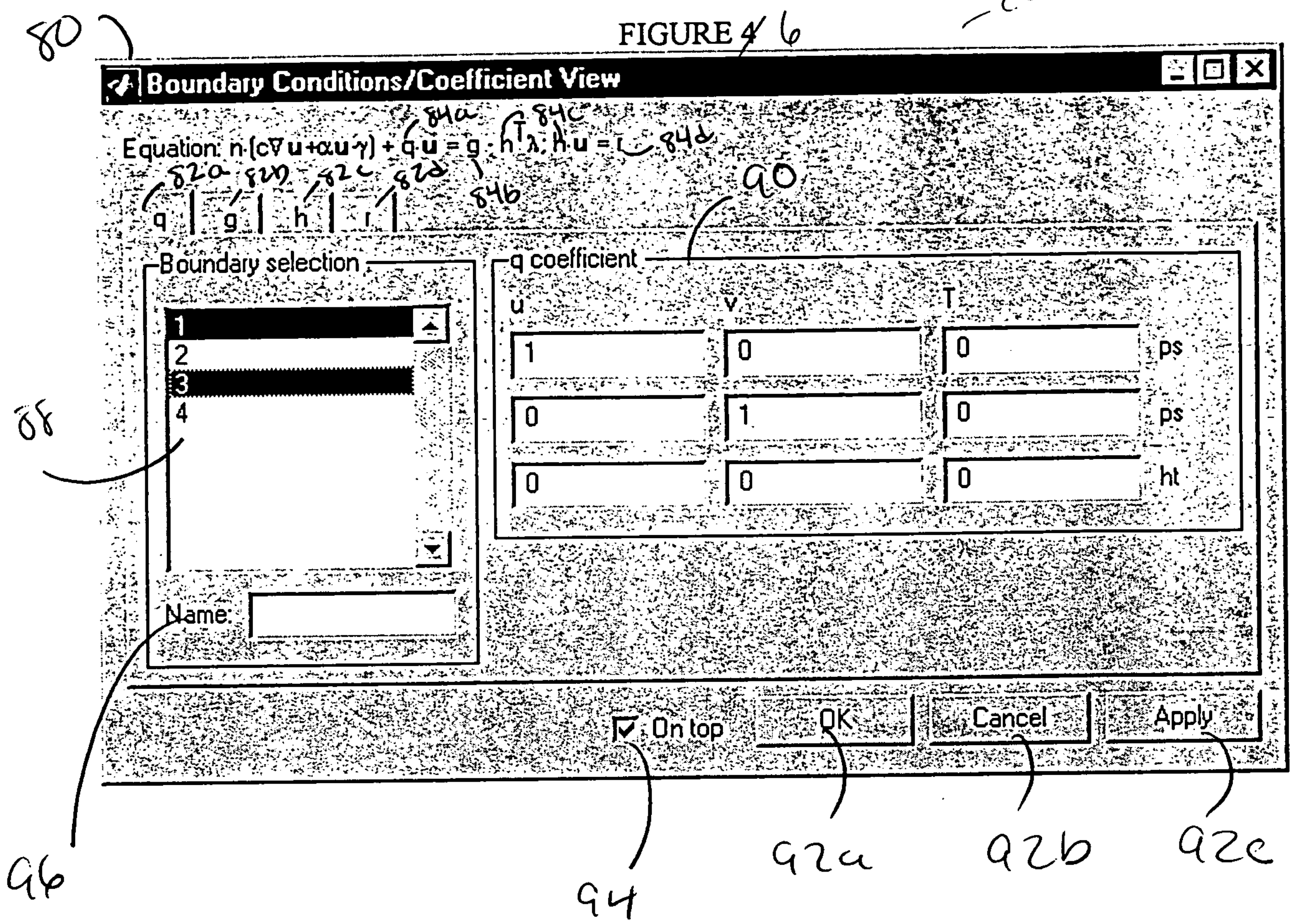
FIGURE ~~3~~ 5



70
Boundary Conditions/ht
Equation: $T = T_0$
Boundary selection
1
2
3
4
5
6
7
Name: 1
Enable borders
72
72b
72a
Boundary coefficients
Unlock
74a
Quantity
Value
Description
q
0
Heat flux
h
0
Heat transfer coefficient
T_{inf}
0
External temperature
C
0
Problem-dependent constant
T_{amb}
0
Ambient temperature
n·(k·gradT)=0
Insulation/symmetry
T
300
Temperature
T=0
Zero temperature
74
On top
OK
Cancel
Apply
74b

FIGURE ~~4~~ 6

— continued



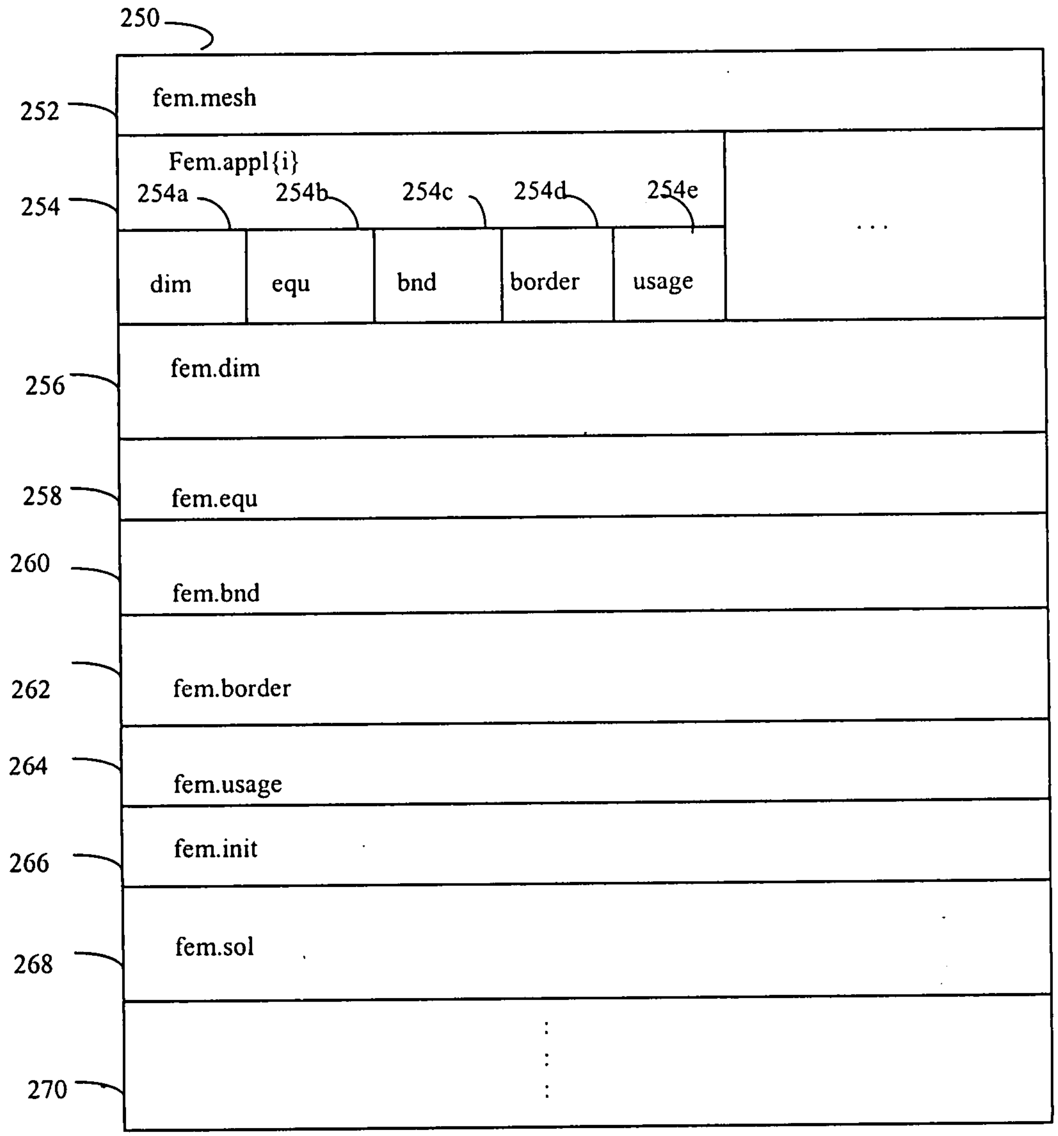

250
252
fem.mesh
254
Fem.appl{i}
254a
254b
254c
254d
254e
dim
equ
bnd
border
usage
...
256
fem.dim
258
fem.equ
260
fem.bnd
262
fem.border
264
fem.usage
266
fem.init
268
fem.sol
270
:
:
:

FIGURE 6A

FIGURE 5/7

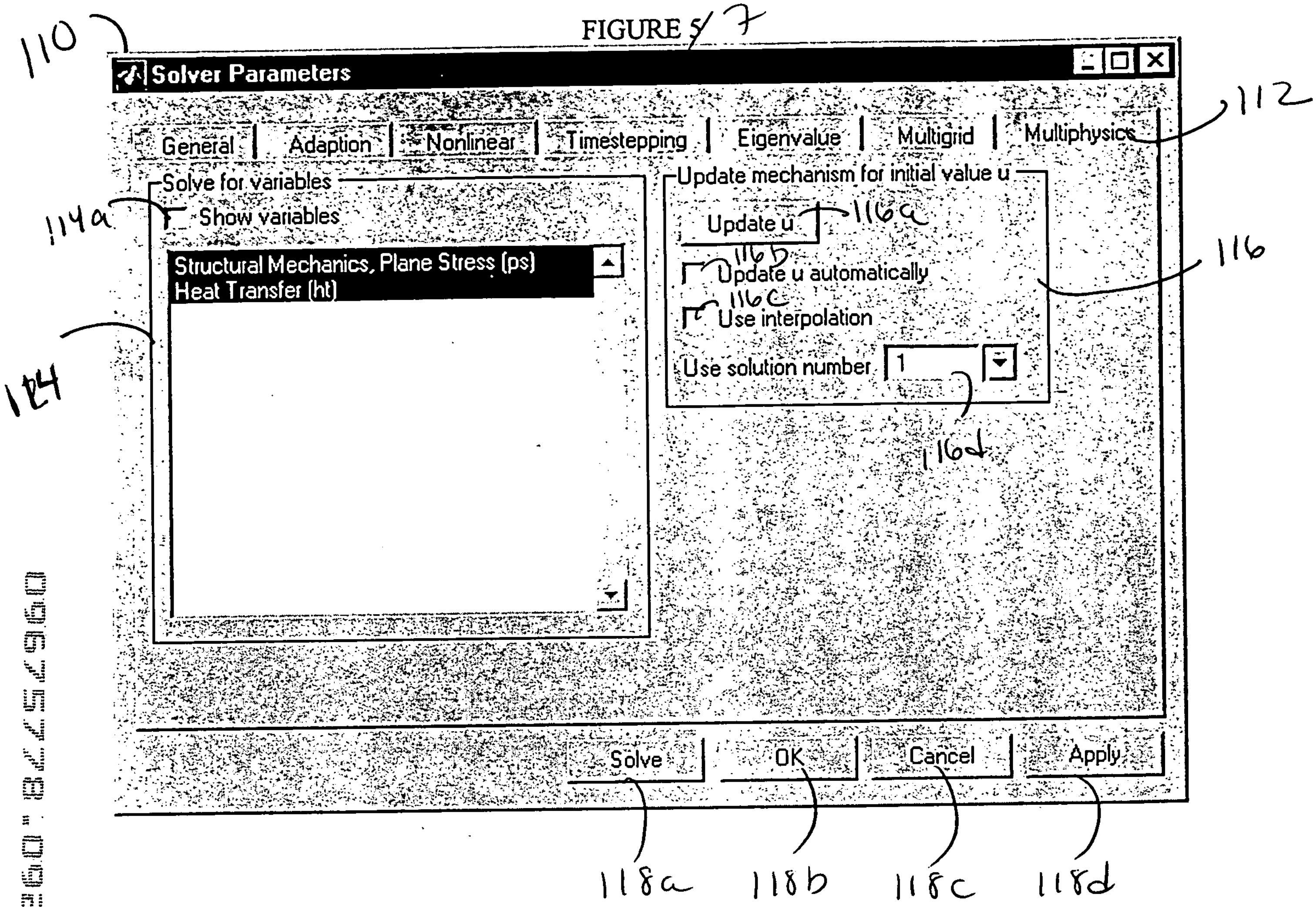

110
Solver Parameters
General
Adaption
Nonlinear
Timestepping
Eigenvalue
Multigrid
Multiphysics
112
Solve for variables
114a
Show variables
Structural Mechanics, Plane Stress (ps)
Heat Transfer (ht)
124
Update mechanism for initial value u
Update u
116a
116b
Update u automatically
116c
Use interpolation
116
Use solution number
1
116d
Solve
OK
Cancel
Apply
118a
118b
118c
118d

FIGURE ~~6~~ 8

$$140\left\{\begin{array}{ll} d_{a\,lk}\dfrac{\partial u_k}{\partial t} - \dfrac{\partial}{\partial x_j}\left(c_{lkji}\dfrac{\partial u_k}{\partial x_i} + \alpha_{lkj}u_k - \gamma_{lj}\right) + \beta_{lki}\dfrac{\partial u_k}{\partial x_i} + a_{lk}u_k = f_l & \Omega \;\; 142 \\ n_j\left(c_{lkji}\dfrac{\partial u_k}{\partial x_i} + \alpha_{lkj}u_k - \gamma_{lj}\right) + q_{lk}u_k = g_l - h_{ml}\lambda_m & \partial\Omega \;\; 146a \\ h_{ml}u_l = r_m & \partial\Omega \;\; 146b \end{array}\right. \quad 146$$

FIGURE ~~7~~ 9

$$150\left\{\begin{array}{ll} d_{a\,lk}\dfrac{\partial u_k}{\partial t} + \dfrac{\partial \Gamma_{lj}}{\partial x_j} = F_l & \Omega \;\; 152 \\ -n_j\Gamma_{lj} = G_l + \dfrac{\partial R_m}{\partial u_l}\lambda_m & \partial\Omega \;\; 154a \\ 0 = R_m & \partial\Omega \;\; 154b \end{array}\right. \quad 154$$

FIGURE 10

$$
324 \left\{
\begin{array}{ll}
\gamma_{lj} = \Gamma_{lj} & f_l = F_l \\[2ex]
c_{lkji} = -\dfrac{\partial \Gamma_{lj}}{\partial\left(\dfrac{\partial u_k}{\partial x_i}\right)} & \alpha_{lkj} = -\dfrac{\partial \Gamma_{lj}}{\partial u_k} \\[4ex]
\beta_{lki} = -\dfrac{\partial F_l}{\partial\left(\dfrac{\partial u_k}{\partial x_i}\right)} & a_{lk} = -\dfrac{\partial F_l}{\partial u_k} \\[4ex]
g_l = G_l & r_l = R_l \\[2ex]
q_{lk} = -\dfrac{\partial G_l}{\partial u_k} & h_{lk} = -\dfrac{\partial R_l}{\partial u_k}
\end{array}
\right.
$$

FIGURE ~~8~~ 11

240

$$\begin{cases} \Gamma_{lj} = -c_{lkji}\dfrac{\partial u_k}{\partial x_i} - \alpha_{lkj}u_k + \gamma_{lj} \\ F_l = f_l - \beta_{lki}\dfrac{\partial u_k}{\partial x_i} - a_{lk}u_k \\ G_l = g_l - q_{lk}u_k \\ R_m = r_m - h_{ml}u_l \end{cases}$$

FIG 12

$$
300\left\{
\begin{aligned}
&\int_\Omega \left(\left(c_{lkji} \frac{\partial u_k}{\partial x_i} + \alpha_{lkj} u_k \right) \frac{\partial v}{\partial x_j} + \left(d_{a\,lk} \frac{\partial u_k}{\partial t} + \beta_{lki} \frac{\partial u_k}{\partial x_i} + a_{lk} u_k \right) v \right) dx + \\
&\int_{\partial\Omega} q_{lk} u_k v\, ds = \int_\Omega \left(\gamma_{lj} \frac{\partial v}{\partial x_j} + f_l v \right) dx + \int_{\partial\Omega} (g_l - h_{ml} \lambda_m) v\, ds \\
&\int_{\partial\Omega} \mu h_{mk} u_k\, ds = \int_{\partial\Omega} \mu r_m\, ds
\end{aligned}
\right.
$$

FIG 13

$$302\begin{cases} \int_{\Omega} \left(\Gamma_{lj}\frac{\partial v}{\partial x_j} + F_l v - d_{alk}\frac{\partial u_k}{\partial t} v\right) dx + \int_{\partial\Omega}\left(G_l + \frac{\partial R_m}{\partial u_l}\lambda_m\right) v\, ds = 0 \\ \int_{\partial\Omega} R_m \mu\, ds = 0 \end{cases}$$

FIG 14

304 $$U_k(x) = \sum_{I=1}^{N_p} U_{I,k}\,\phi_I(x), \qquad \Lambda_m(x) = \sum_{K=1}^{N_e} \sum_{L=1}^{n} \Lambda_{K,L,m}\,\Psi_{K,L}(x)$$

FIG 15

$$206\left\{\begin{aligned}
&\int_{\tau}\left(c_{lkji}U_{I,k}\frac{\partial\phi_I}{\partial x_i}+\alpha_{lkj}U_{I,k}\phi_I\right)\frac{\partial\phi_J}{\partial x_j}dx+\\
&\int_{\tau}\left(d_{a\ lk}\frac{\partial U_{I,k}}{\partial t}\phi_I+\beta_{lki}U_{I,k}\frac{\partial\phi_I}{\partial x_i}+a_{lk}U_{I,k}\phi_I\right)\phi_J dx+\\
&\int_{\partial\tau}q_{lk}U_{I,k}\phi_I\phi_J ds = \int_{\tau}\left(\gamma_{lj}\frac{\partial\phi_J}{\partial x_j}+f_l\phi_J\right)dx+\\
&\int_{\partial\tau}(g_l-h_{ml}\Lambda_{K,L,m}\Psi_{K,L})\phi_J ds
\end{aligned}\right.$$

FIG 16

$$308 \left\{ \int_{\partial\tau} h_{mk} U_{I,k} \phi_I \Psi_{K,L} \, ds = \int_{\partial\tau} r_m \Psi_{K,L} \, ds \right.$$

FIG 17

312 $$\begin{cases} \int_{\tau}\left(\Gamma_{lj}\frac{\partial\phi_J}{\partial x_j}+F_l\phi_J-d_{alk}\frac{\partial u_k}{\partial t}\phi_J\right)\,dx+\int_{\partial\tau}\left(G_l+\frac{\partial R_m}{\partial u_l}\Lambda_{K,L,m}\Psi_{K,L}\right)\phi_J\,ds=0 \\ \int_{\partial\tau}R_m\Psi_{K,L}\,ds=0 \end{cases}$$

FIG 18

310

$$DA_{(J,I),(I,k)} = \int_\tau d_{a\,lk} \phi_I \phi_J \, dx$$

$$C_{(J,I),(I,k)} = \int_\tau c_{lkji} \frac{\partial \phi_I}{\partial x_i} ? \frac{\partial \phi_J}{\partial x_j} \, dx$$

$$AL_{(J,I),(I,k)} = \int_\tau \alpha_{lkj} \phi_I ? \frac{\partial \phi_J}{\partial x_j} \, dx$$

$$BE_{(J,I),(I,k)} = \int_\tau \beta_{lki} \frac{\partial \phi_I}{\partial x_i} \phi_J \, dx$$

$$A_{(J,I),(I,k)} = \int_\tau a_{lk} \phi_I \phi_J \, dx$$

$$Q_{(J,I),(I,k)} = \int_{\partial\tau} q_{lk} \phi_I \phi_J \, ds$$

$$GA_{(J,l)} = \int_\tau \gamma_{lj} \frac{\partial \phi_J}{\partial x_j} \, dx$$

$$F_{(J,l)} = \int_\tau f_l \phi_J \, dx$$

$$G_{(J,l)} = \int_{\partial\tau} g_l \phi_J \, ds$$

$$H_{(K,L,m),(I,k)} = \int_{\partial\tau} h_{mk} \phi_I \Psi_{K,L} \, ds$$

$$R_{(K,L,m)} = \int_{\partial\tau} r_m \Psi_{K,L} \, ds$$

FIG 19

320

$$\begin{cases} DA\frac{\partial U}{\partial t} + (C + AL + BE + A + Q)U + H^T\Lambda = GA + F + G \\ HU = R \end{cases}$$

FIG 20

$$322\left\{\begin{array}{l} DA\frac{\partial U}{\partial t} + H^{T}\Lambda = GA + F + G \\ R = 0 \end{array}\right.$$

FIG 21

$$324 \begin{cases} J(U^{(k)})\Delta U^{(k)} = -\rho(U^{(k)}) \\ U^{(k+1)} = U^{(k)} + \lambda_k \Delta U^{(k)} \end{cases}$$

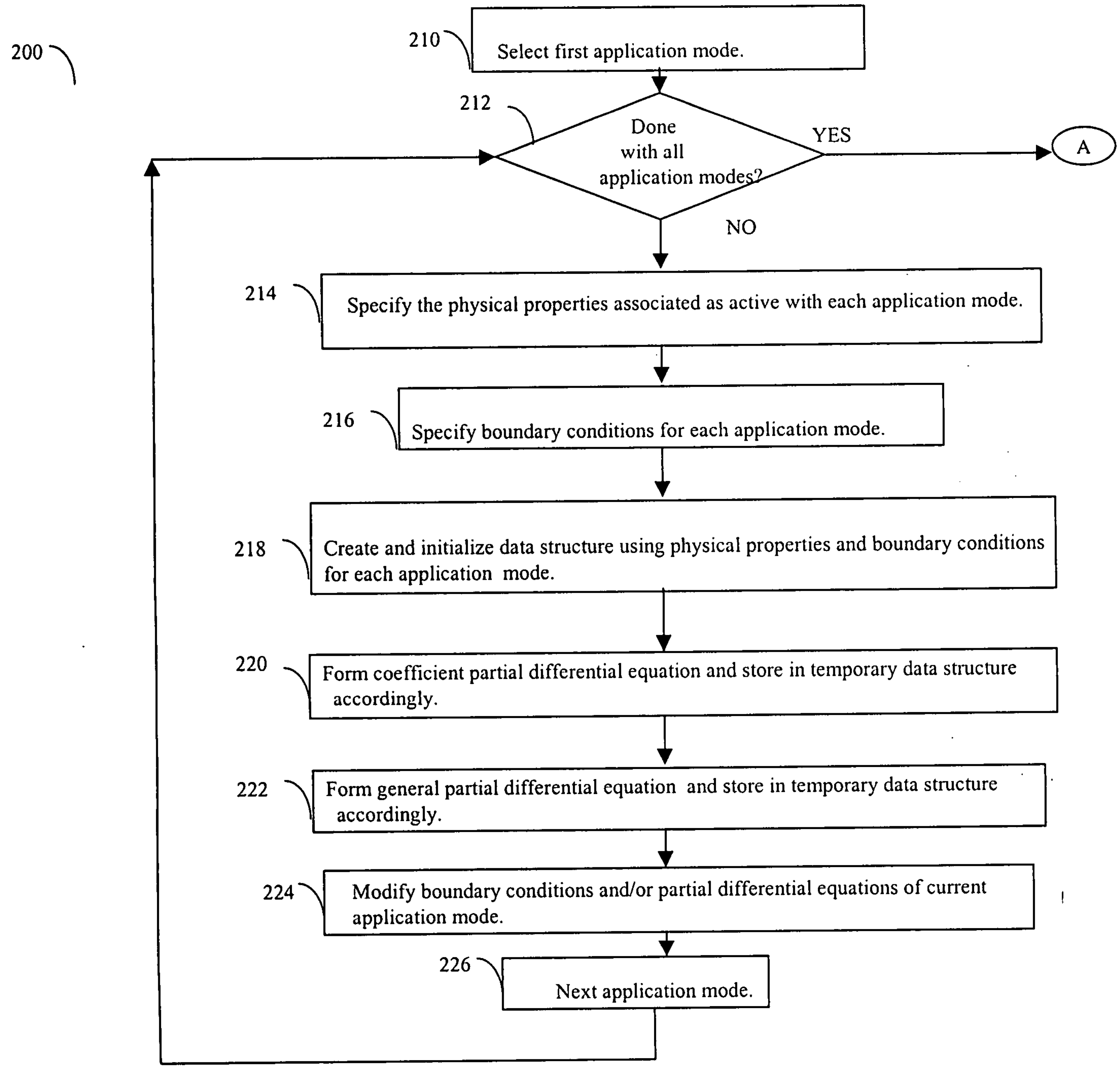

200
210
Select first application mode.
212
Done with all application modes?
YES
A
NO
214
Specify the physical properties associated as active with each application mode.
216
Specify boundary conditions for each application mode.
218
Create and initialize data structure using physical properties and boundary conditions for each application mode.
220
Form coefficient partial differential equation and store in temporary data structure accordingly.
222
Form general partial differential equation and store in temporary data structure accordingly.
224
Modify boundary conditions and/or partial differential equations of current application mode.
226
Next application mode.

FIGURE 22

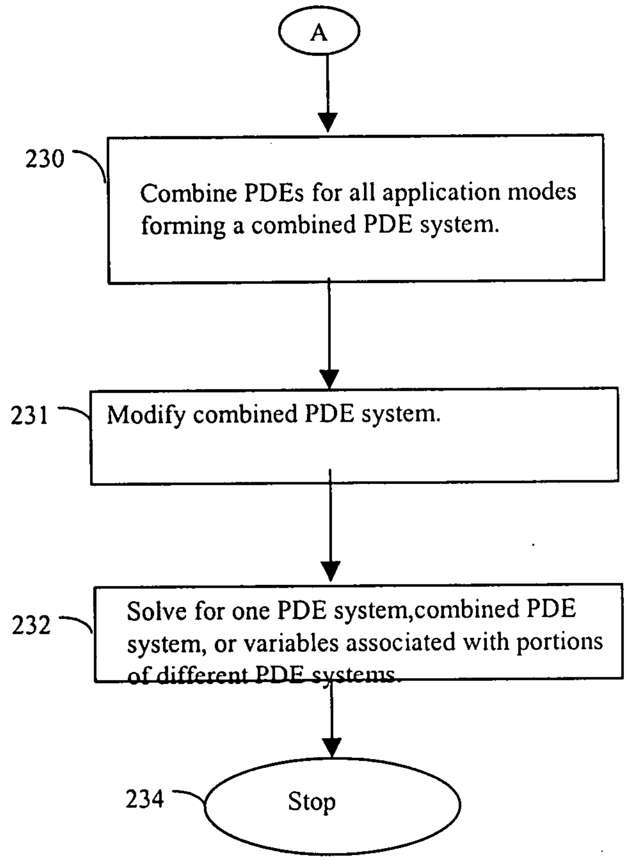
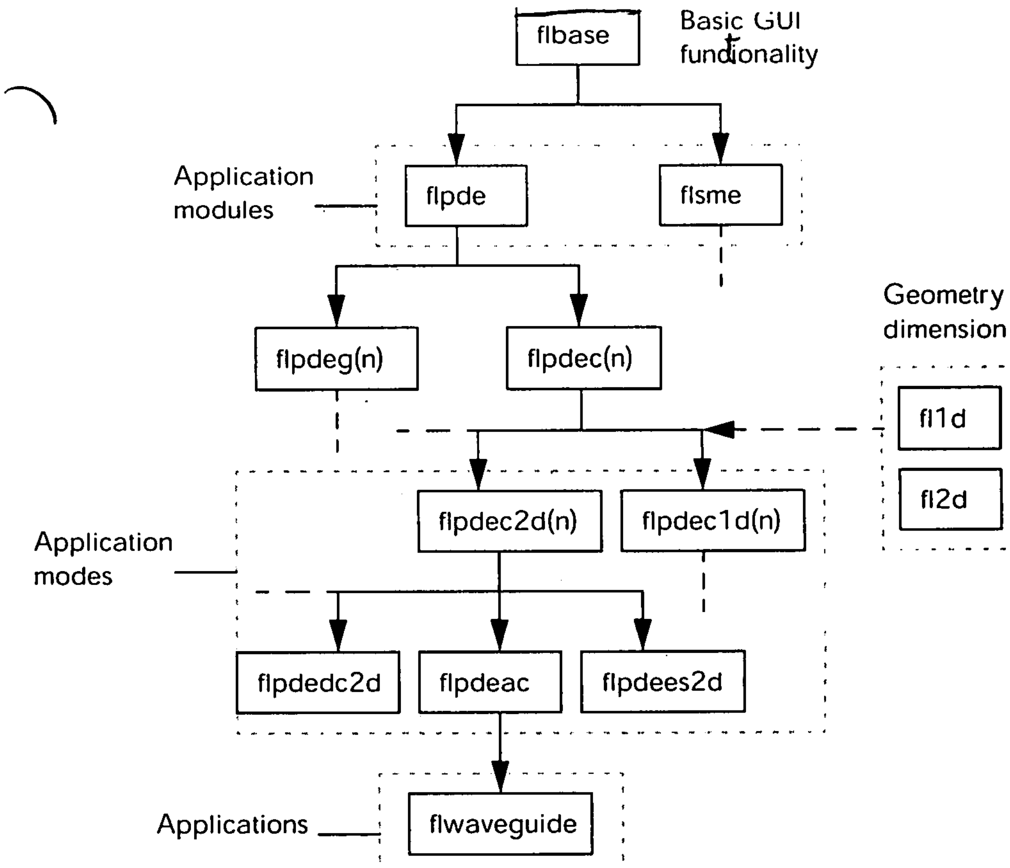

A
230
Combine PDEs for all application modes forming a combined PDE system.
231
Modify combined PDE system.
232
Solve for one PDE system,combined PDE system, or variables associated with portions of different PDE systems.
234
Stop

FIGURE 23

006260 " 8225960

500



The class hierarchy of FEMLAB

FIGURE 24

1-D Physics Application Modes

Application mode	Class name	Parent class
Diffusion	`flpdedf1d`	`flpdedf`
Heat Transfer	`flpdeht1d`	`flpdeht`

502

: **1-D PDE Application Modes**

Application mode	Class name	Parent class
Coefficient PDE model, n variables	`flpdec1d`*(n)*	`flpdec`*(n)*
General PDE model, n variables	`flpdeg1d`*(n)*	`flpdeg`*(n)*

504

FIGURE 25

2-D Physics Application Modes

Application mode	Class name	Parent class
AC Power Electromagnetics	flpdeac	flpdec2d
Conductive Media DC	flpdedc2d	flpdedc
Diffusion	flpdedf2d	flpdedf
Electrostatics	flpdees2d	flpdees
Magnetostatics	flpdems2d	flpdems
Heat Transfer	flpdeht2d	flpdeht
Incompressible Navier-Stokes	flpdens2d	flpdens
Structural Mechanics, Plane Stress	flpdeps	flpdec2d
Structural Mechanics, Plane Strain	flpdepn	flpdec2d

506

PDE Application Modes

Application mode	Class name	Parent class
Coefficient PDE model, n variables	flpdec2d*(n)*	flpdec*(n)*
General PDE model, n variables	flpdeg2d*(n)*	flpdeg*(n)*

508

FIGURE 26

FIGURE 27

Application Object Properties

Property name	Description	Data type
`dim`	Names of the dependent variables	Cell array of strings
`form`	PDE form	String (`coefficient`/`general`)
`name`	Application name	String
`parent`	Parent class names	String, cell array of strings, or the empty matrix
`sdim`	Names of the independent variables (space dimensions)	Cell array of strings
`submode`	Name of current submode	String (`std`/`wave`)
`tdiff`	Time differentiation flag	String (`on`/`off`)

570

512

```
function obj = myapp()
%MYAPP Constructor for a FEMLAB application object.

obj.name = 'My first FEMLAB application';
obj.parent = 'flpdeht2d';

% MYAPP is a subclass of FLPDEHT2D:
p1 = flpdeht2d;
obj = class(obj,'myapp',p1);
set(obj,'dim',default_dim(obj));
```

FIGURE 28

Physics Modeling Methods

514

Function	Purpose
`appspec`	Return application specifications.
`bnd_compute`	Convert application-dependent boundary conditions to generic boundary coefficients.
`default_bnd`	Default boundary conditions.
`default_dim`	Default names of dependent variables.
`default_equ`	Default PDE coefficients/Material parameters.
`default_init`	Default initial conditions.
`default_sdim`	Default space dimension variables.
`default_var`	Default application scalar variables.
`dim_compute`	Return dependent variables for an application.
`equ_compute`	Convert application-dependent material parameters to generic PDE coefficients.
`form_compute`	Return PDE form.
`init_compute`	Convert application-dependent initial conditions to generic initial conditions.
`posttable`	Define assigned variable names and post-processing information.

FIGURE 29

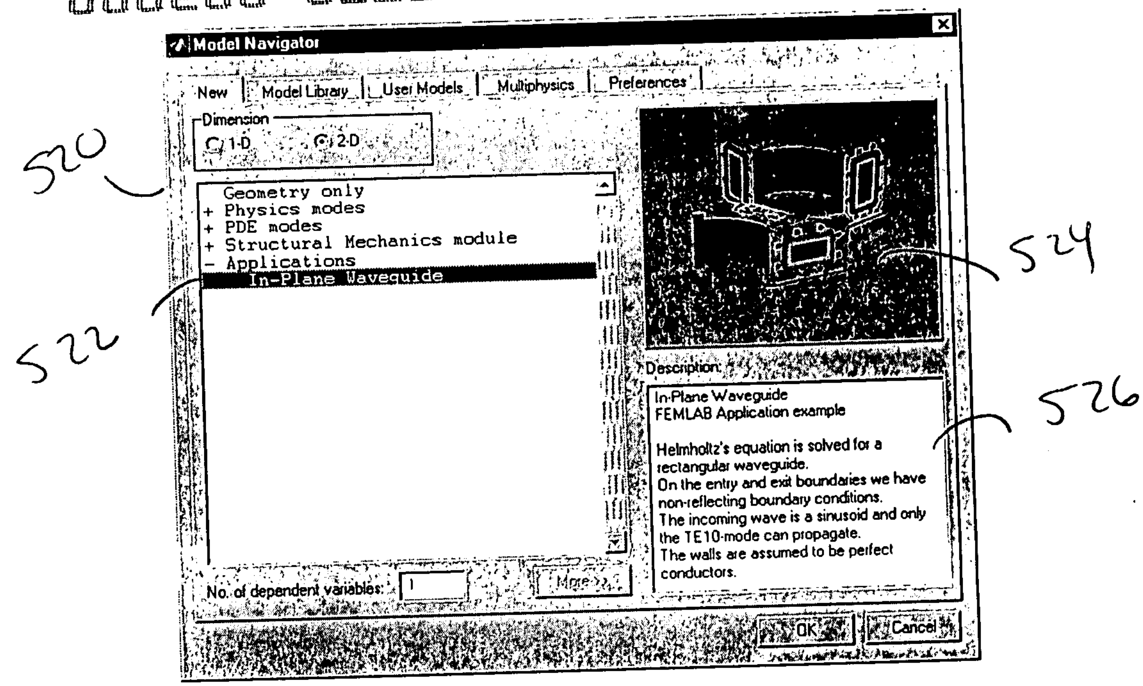

Model Navigator
New
Model Library
User Models
Multiphysics
Preferences
Dimension
1-D
2-D
Geometry only
+ Physics modes
+ PDE modes
+ Structural Mechanics module
- Applications
In-Plane Waveguide
No. of dependent variables:
1
More >>
Description:
In-Plane Waveguide
FEMLAB Application example
Helmholtz's equation is solved for a rectangular waveguide.
On the entry and exit boundaries we have non-reflecting boundary conditions.
The incoming wave is a sinusoid and only the TE10-mode can propagate.
The walls are assumed to be perfect conductors.
OK
Cancel
520
522
524
526

FIGURE 30

530 $$\Delta E_z + (2\pi i k)^2 E_z = 0$$

532 $$k = \frac{1}{\lambda} = \frac{f}{c}$$

534 $$\bar{n} \cdot (\nabla E_z) + 2\pi i k_x E_z = 4\pi i k_x \sin\left(\frac{\pi}{d}(y - y_0)\right)$$

536 $$k^2 = k_x^2 + k_y^2$$

538 $$k_x = \sqrt{\frac{1}{\lambda^2} - \frac{1}{(2d)^2}}$$

540 $$n \cdot (\nabla E_z) + 2\pi i k_x E_z = 0$$

542 $$E_z = 0$$

544 $$f_c = \frac{c}{2d}$$

FIGURE 31

```
function obj = flwaveguide(varargin)
%FLWAVEGUIDE Constructor for a Waveguide application object.

obj.name = 'In-Plane Waveguide';
obj.parent = 'flpdeac';

% FLWAVEGUIDE is a subclass of FLPDEAC:
p1 = flpdeac;
obj = class(obj,'flwaveguide',p1);
set(obj,'dim',default_dim(obj));
```

550

FIGURE 32

552

fem.user fields

Field	Description
geomparam	1-by-2 structure of geometry parameters.
entrybnd	Index to the entry boundary.
exitbnd	Index to the exit boundary.
freqs	Frequency vector

FIGURE 33

fem.user fields

554

Field	Description
`startpt`	Index of the lower left corner point of the waveguide.
`type`	Type of waveguide. ('*straight* or '*elbow*)

FIGURE 34

geomparam fields

556

Field	Description	Defaults for elbow	Defaults for straight
`entrylength`	Length of the entrance part of the waveguide.	0.1	0.1
`exitlength`	Length of the exit part of the waveguide.	0.1	Not used
`radius`	Outer radius of the waveguide bend.	0.05	Not used
`width`	Width of the waveguide.	0.025	0.025
`cavityflag`	Turn resonance cavity *on* or *off*.	0	0
`cavitywidth`	Width of the resonance cavity.	0.025	0.025
`postwidth`	Width of the protruding posts.	0.005	0.005
`postdepth`	Depth of the protruding posts.	0.005	0.005

FIGURE 35